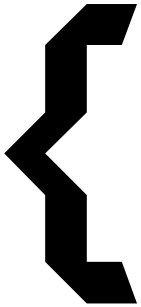
Google Protocol Buffers

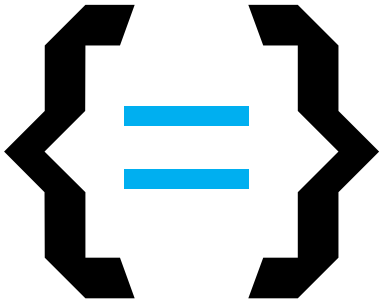
Pavel Janečka

únor 2014



Google Protocol Buffers

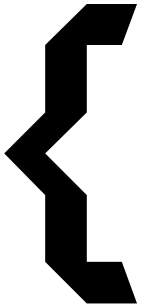
GDG Hradec Králové



Univerzita Hradec Králové
Fakulta informatiky a managementu

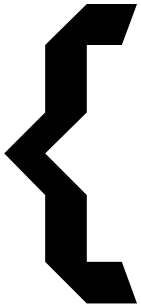
/Sorceror

+PavelJanečka



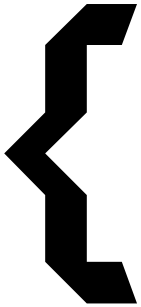
Adresář

- Java serializace
  - Znemožňuje změnu serializovaných tříd
  - Zvyšuje pravděpodobnost bugů
  - Jazykově nepřenositelná
- Vlastní formát
  - Jednoduché a flexibilní
  - Nutnost psaní encoderu a parseru
  - Nejlepší řešení pro jednoduchá data



Adresář

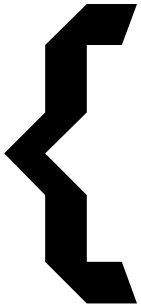
- **XML**
  - “Human-readable”
  - Dostupné knihovny
  - Příliš ukecané
    - Náročnost na paměť a diskový prostor
  - Dopad na výkon aplikace



Adresář

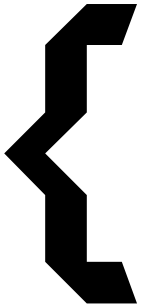
- Google Protocol Buffers
  - Automatizované řešení implicitního formátu
  - Jazykově a platformě neutrální
  - Oproti XML
    - Jednodušší zápis
    - Menší (3x – 10x), rychlejší (20x – 100x)
    - Méně víceznačné
    - Jednodušeji využitelné programově

http://stackoverflow.com/questions/647779/high-performance-serialization-java-vs-google-protocol-buffers-vs



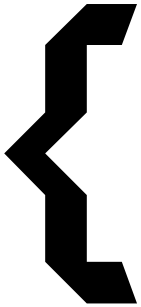
Historie

- Interně využívané v Google
  - Verze 1 – rok 2001
  - Verze 2 – rok 2008
    - Apache 2.0 licence
  - 48 tisíc zpráv, 12 tisíc *.proto souborů
    - Komunikační protokoly
    - RPC
    - Perzistence dat



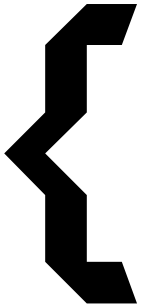
Zpráva ~ message

- Definuje strukturu dat ~ *.proto soubor
  - Jeden soubor může obsahovat více zpráv
- Ekvivalent třídy / struktury
- Položky ~ skaláry, výčtové typy, zprávy
  - Unikátní číselný tag
- Rozšíření (extensions)
  - Nejedná se o dědičnost
- Řádkové komentáře ~ // comment



Zpráva ~ pravidla výskytu

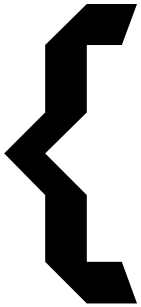
- Každá položka musí mít pravidlo výskytu
  - required – právě jedna hodnota
    - Pokud není položka vyplněna zprávu nelze vytvořit
    - Při deserializaci je zpráva bez povinných hodnot automaticky zahozena
    - **Jednou** required **navždy** required
  - optional – jedna nebo žádná hodnota
  - repeated – žádná nebo více hodnot



Zpráva ~ datové typy

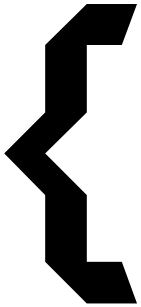
- Skalární typy

*.proto	Java	C++
double	double	double
float	float	float
(u/s/fixed)int32	int	(u)int32
(u/s/fixed)int64	long	(u)int64
bool	boolean	bool
string	String	string
bytes	ByteString	string



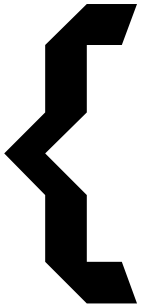
Zpráva ~ datové typy

- Výčtový typ ~ enum
  - Skupina konstant a jejich hodnot
  - Může i nemusí být součástí konkrétní zprávy
    - Rozdíl v dotazovaní na hodnoty
- Zpráva ~ message
  - Součástí zprávy může být i jiná zpráva
  - Typem je pak jméno samotné zprávy
  - Mohou i nemusí být vnořené



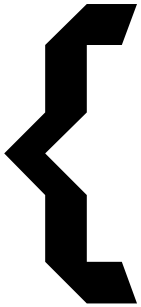
Zpráva ~ message

- Balíček ~ package
  - Jednotlivé *.proto soubory mohou obsahovat definice balíčků ~ namespace
  - Typ je pak nutné definovat včetně jména uvedeného balíčku
- Služby ~ service
  - GPB lze využít i k RPC (Remote Procedure Call)
  - 3rd party nástroje



Zpráva ~ message

- Nastavení ~ option
  - Dodatečná definice vlastností zpráv a položek
  - Zprávy
    - java_outer_classname ~ jméno generované třídy
    - java_package ~ balíček generované třídy
    - optimize_for ~ SPEED, CODE_SIZE, LITE_RUNTIME
  - Položky
    - packed ~ optimalizace repeated číselných hodnot
    - deprecated ~ generuje @Deprecated anotaci

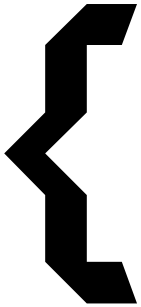


Live coding challenge

AddressBookProto

GDG Hradec Králové

Univerzita Hradec Králové
Fakulta informatiky a managementu

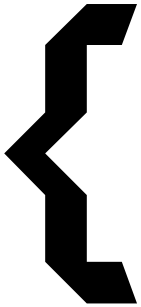


Style guide

- Jazyková i platformní neutrálnost
- Konvence zápisu
  - Zpráva ~ CamelCase NázevZprávy
  - Položka ~ malá_písmena_s_podtržítky
  - Výčtový typ ~ KAPITÁLKY_S_PODTRŽÍTKY
- Java generovaný kód
  - set a get metody pro položky zprávy
  - set metody vrací instanci Builder třídy

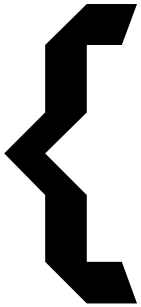
GDG Hradec Králové

Univerzita Hradec Králové
Fakulta informatiky a managementu



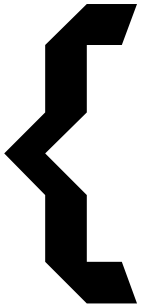
Vnitřní struktura a kódování

- Založeno na varint kódování
  - Kromě posledního bajtu mají všechny nastavený most significant bit
  - wire-type ~ definuje délky čtené hodnoty
  - Řetězce ~ hlavička + UTF-8
- Zachovává pořadí hodnot v repeated položkách, ale nezachovává pořadí položek ve zprávě



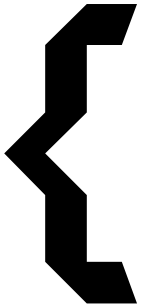
Generovaný kód ~ Java API

- Kompilace *.proto zprávy
  - protoc --java_out=./gen foo.proto
  - Generování ovlivněno nastaveními ~ options
- Message ~ instance jsou immutable
  - static Foo parseFrom(...)
  - Foo.Builder newBuilder([Foo prototype])
- Builder ~ slouží ke tvorbě zpráv
  - Setter metody vrací Builder instanci ~ řetězení



Generovaný kód – Java API

- Jednoduché položky ~ required, optional
  - bool hasFoo(), int getFoo()
  - Builder setFoo(int value), Builder clearFoo()
- Vícenásobné položky ~ repeated
  - int getFooCount(), int getFoo(int index)
  - List<Integer> getFooList()
  - Builder setFoo(int index, int value)
  - Builder addFoo(int value)
  - Builder addAllFoo(List<Integer> value)
  - Builder clearFoo()



Generovaný kód – Java API

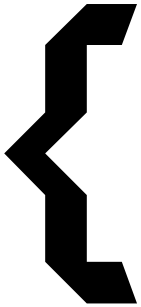
- Mnoho dalších generovaných metod
  - Podpora reflexe
    - isInitialized() ~ vrací true pokud je položka nastavena
    - toString() – human-readable řetězec včetně obsahu (debug)
    - mergeFrom(Message other) – sloučí obsah zpráv

Serializace

byte[] **toByteArray**()
void **writeTo**(OutputStream output)

Deserializace

static Foo **parseFrom**(byte[] data)
static Foo **parseFrom**(InputStream input)

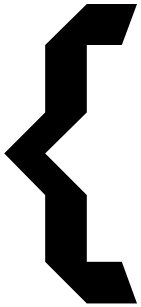


Live coding challenge

AddressBookApp

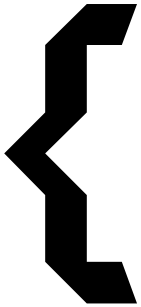
GDG Hradec Králové

Univerzita Hradec Králové
Fakulta informatiky a managementu



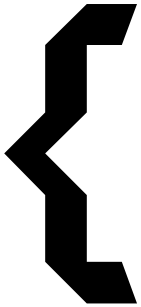
Best practices

- OO design
  - Generované třídy jsou pouze “dumb data holders” a neměli by být používány v kódu
    - Vytvořte si vlastní wrappery
  - Generované třídy nejsou vhodní kandidáti pro dědičnost
    - Kód se mění, špatný návrh



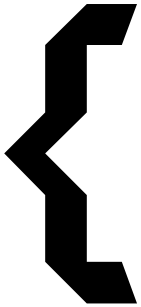
Best practices

- Aktualizace *.proto souborů
  - Neměňte číselné tagy jednotlivých položek
    - OBSOLETE_ prefix
  - Nepřidávejte či odebírejte required položky
    - Zprávy vytvořené „starým“ kódem budou automaticky zahozeny
  - Lze smazat optional a repeated položky, či je nově přidat
    - Musí mít jedinečný číselný tag



Best practices

- Streamování více zpráv
  - Před každou zprávu zaznamenat velikost
  - CodedInputStream
- Velké datasety
  - Vhodnější pro větší množství malých zpráv
- Sebe popisující zpráva
  - FileDescriptorSet parametr kompileru
  - DynamicMessage třída



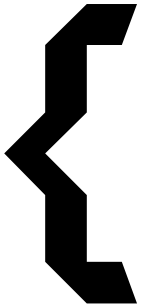
Best practices

- Hodnota null
  - Nelze nastavit v GBP, nutno obcházet přes vlastní definici null hodnoty
    - hasFoo() a clearFoo() metody
    - Některé jazyky neobsahují null hodnotu
    - Bezpečnostní riziko ~ null exploit
    - Problémy s řetězením

https://code.google.com/p/protobuf/issues/detail?id=57#c40

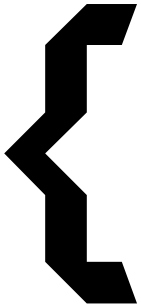
http://www.codedmike.com/java-nulls-in-protobuf/

http://stackoverflow.com/questions/9184215/whats-the-preferred-way-to-encode-a-nullable-field-in-protobuf



Add-ons

- Implementace v různých jazycích
  - Java, C++, Python ~ Google
  - AS, C, C#, Closure, D, Erlang, Haskell, Go, JS, Lua, Matlab, Perl, PHP, Prolog, R, Ruby, Scala
- Množství 3rd party RPC implementací
  - C/C++, Java, C#, Python
- Ostatní nástroje
  - IDE pluginy, JSON, WireShark packet sniffer



Závěr

- **„Pouze“** serializace dat
  - Programově lépe ovladatelné
  - Rychlejší a objemově menší než XML
  - Nutnost rekompilace tříd při změně definic
    - Maven plugin
    - Při rozumných změnách zpětná kompatibilita
- Google technologie
- Více jak 10 let vývoje